Assessing Physics Informed Neural Networks in Ocean Modelling and Climate Change Applications

Taco de Wolff, Hugo Carrillo, Luis Martí, Nayat Sanchez-Pi

Assessing Physics Informed Neural Networks in Ocean Modelling and Climate Change Applications

Taco de Wolff, Hugo Carrillo, Luis Martí & Nayat Sánchez-Pi
Inria Chile Research Center
Av. Apoquindo 2827, piso 12, Las Condes, Santiago, Chile
{taco.dewolff,hugo.carrillo,lmarti,nayat.sanchez-pi}@inria.cl

Abstract

The carbon pump of the world's oceans plays a vital role in the biosphere and climate of the earth, urging improved understanding of the functions and influences of the oceans for climate change analyses. State-of-the-art techniques are required to develop models that can capture the complexity of ocean currents and temperature flows. We will explore the benefits of using physics informed neural networks (PINNs) for solving partial differential equations related to ocean modeling; such as the wave, shallow water, and advection-diffusion equations. PINNs account for the deviation from physical laws in order to improve learning and generalization. However, in this work, we observe worse training and generalization results, possibly due the amount of data used in training.

1 Introduction

The oceans play a key role in the biosphere (Sánchez-Pi et al., 2020). They regulate the carbon cycle by absorbing emitted CO_2 through the biological pump and dissipate a large part of the heat that is retained in the atmosphere by the remaining CO_2 and other greenhouse gases. The biological pump is driven by photosynthetic microalgae, herbivores, and decomposing bacteria. Understanding the drivers of micro- and macroorganisms in the ocean is of paramount importance to understand the functioning of ecosystems and the efficiency of the biological pump in sequestering carbon and thus abating climate change.

Similarly, there is also a clear scientific consensus about the effects of climate change on the oceans. Changes like the shift in temperatures, an increase of acidification, deoxygenation of water masses, perturbations in nutrient availability and biomass productivity, to mention a few, have a dramatic impact on almost all forms of life in the ocean with further consequences on food security, ecosystem services, and the well-being of coastal communities and humankind in general.

Consequently, we end up with a cyclic dependency: to understand and create policies to mitigate climate change it is of primordial importance to understand the ocean and to understand the ocean it is necessary to understand how climate change is impacting it. Therefore, this is not only an urgent but also a scientifically demanding task that must be addressed with a scientific cohort approach involving different scientific areas: state-of-the-art artificial intelligence, machine learning, applied math, modelling, and simulation, and, of course, marine biology and oceanography.

That is why it can be hypothesized that it is necessary to develop new state-of-the-art artificial intelligence, machine learning, and mathematical modelling tools that will enable us to move forward with the understanding of our oceans and to understand, predict and hopefully mitigate the consequences of climate change.

Modelling oceanic fluid dynamics is essential but also very expensive in the computational sense. Mathematical models like the Navier-Stokes equations (Temam, 2001) can express most of the processes of interest (see McLean (2012)). However, the high computational cost involved in applying them with the correct precision level makes their application intractable. Machine learning (ML) and, in particular, neural networks are a competitive alternative. However, these approaches in general require a large amount of data to serve as examples for the learning process. In ocean research, obtaining data is challenging and expensive. The approach of hybrid models seems to be a good alternative to tackle this problem.

Physics informed neural networks (PINNs) as proposed by (Raissi et al., 2019) are a hybrid approach that takes into account an ML and mechanistic model, which are two different paradigms. We aim to answer the following question: *what is the importance and relevance of the physical component in PINNs in improving the predictive power of a neural network?* We aim to find an efficient configuration of parameters with numerical experiments for the prediction of data governed by the wave equation and the shallow water and advection-diffusion equations.

The hyperbolic nature of the wave equation is a useful start for investigating typical hyperbolic ocean modelling equations such as the shallow water equations. It is possible to reduce the shallow water equations to the wave equation under certain suitable conditions, see for example (Stoker, 2011). Furthermore, coupling the shallow water and advection-diffusion equations could bring a simple toy model for the temperature in oceans. Hence our objective is to improve our understanding of ocean behaviour and its temperature flows to better model its interaction with climate change.

This paper deals with the problem of predicting values of physical laws at given points in time. We choose PINNs to investigate this problem, so that we are going to assess the efficacy of PINNs applied to simple models related to the ocean. The rest of this paper is organized as follows. In Section 2 we present a brief rationale about the choice of the PINN technique. In section 3 we provide the reader with the considered model equations. In Section 3 we provide the values of the functions involved in partial differential equation (PDE) simulations and the essential aspects of PINN input generation. In Section 5 we briefly explain the PINN method. Then in Section 6 we analyse with numerical examples the performance of PINNs for the estimation of the solution and parameters of the PDEs. Finally, in Section 7 we discuss the results and propose ways to continue this study.

2 Foundations and context

High-dimensional partial differential equations (PDEs) are a common fixture in areas as diverse as physics, chemistry, engineering, finance, etc. Their enormous success has been hampered by its limited computational viability. Numerical methods for solving PDEs such as finite difference or finite element methods become infeasible in higher dimensions due to the explosion in the number of grid points and the demand for reduced time step sizes.

Addressing PDE scalability by approximating them by neural networks has been considered in various forms either by creating a training data set by directly sampling from an *a priori* fixed mesh or by directly sampling the solution in a random manner (see Sirignano & Spiliopoulos (2018) for an overview). Consequently, a database of synthetic data coming from the numerical resolution of PDE-based models is generated on a broad range of scenarios. These data sets are then used to train and validate deep neural networks as in (Dupont et al., 2019; Chen et al., 2018).

These approaches have the limitation of high computational costs for creating such data sets. To subvert the high computational costs, PINNs were recently proposed by (Raissi et al., 2019) as a hybrid approach that considers a process where a source of physical knowledge in the form of PDEs is available. They specify the problem of training the neural networks as a multi-objective learning task, where we want to minimize the error with the data as well as the error with a physical law. Now we have three problems instead of one: minimizing the error with the data, minimizing the error with the physical law, and the combination of training for both objectives. (Raissi et al., 2019) solves this by incorporating a physics loss term to the loss function, including a relative weight between both loss terms.

3 Model equations

In the following we present the equations considered in our study and give a brief interpretation of them and the functions involved. They are the wave, shallow water, and advection-diffusion equations. Each of these equations is defined in $\Omega \times \mathbb{R}_0^+$, where $\Omega \subseteq \mathbb{R}^n$ is an open set, and we consider $n = 2$.

3.1 Wave equation

We consider the following wave equation:

$$\frac{\partial^2 \eta}{\partial t^2} - \nabla \cdot (H \nabla \eta) \quad = 0 \tag{1}$$

where $\eta : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ represents the superficial fluctuations of a water container or channel and $H : \overline{\Omega} \to \mathbb{R}_0^+$ is the depth of the water from a reference level.

This equation can be written as

$$f_{waves}[\eta, H] = 0\,. \tag{2}$$

This is where the physics loss will be taken from for this PDE.

3.2 Shallow water equations

The shallow water equations are an approximation of liquid body movements where the horizontal scale is much larger than the vertical scale. It is used to model e.g. floods, tidal currents, and tsunami waves. The standard shallow water system (Saint-Venant, 1871) considering negligible bottom friction and vertical movement of water, as well as constant water pressure and incompressibility, is given by:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\left((H+\eta)u\right) + \frac{\partial}{\partial y}\left((H+\eta)v\right) \quad = \quad 0 \tag{3}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv \quad = \quad -g\frac{\partial \eta}{\partial x} - bu - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{4}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} - fu \quad = \quad -g\frac{\partial \eta}{\partial y} - bv - \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{5}$$

where $u : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ and $v : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ are the height-averaged velocities in the x and y directions respectively, h the height deviation of the surface from its mean height $H : \overline{\Omega} \to \mathbb{R}_0^+$, g the acceleration due to gravity, f the Coriolis coefficient, b the viscous drag coefficient, and ν the kinematic viscosity.

The equations can be further simplified by considering only linear terms and either neglecting the nonlinear terms or by approximating them with a linear function. The linearized version of the shallow water equations is of particular interest because it is very similar to the equations describing acoustic waves. Neglecting the Coriolis effect, advective, and frictions terms, we obtain

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H\boldsymbol{u}) \quad = \quad 0 \text{ and} \tag{6}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + g\nabla\eta \quad = \quad \boldsymbol{\psi} \tag{7}$$

where $\boldsymbol{u} = (u, v)$ and $\boldsymbol{\psi} \in \mathbb{R}^2$ accounts for atmospheric pressure variations and wind stress. We take $g = 9.81\frac{\text{m}}{\text{s}^2}$, $H = 10$m, and $\boldsymbol{\psi} = \mathbf{0}\frac{\text{m}}{\text{s}^2}$. This equation can be written as

$$\boldsymbol{f}_{SW}[u, v, \eta] = \mathbf{0} \tag{8}$$

where $\boldsymbol{f}_{SW} = (f_{SW}^1, f_{SW}^2, f_{SW}^3)$. This is where the physics loss will be taken from for this equation.

3.3 Advection-diffusion equation

We consider the advection-diffusion equation for heat transfer, assuming that the diffusion coefficient is constant, there are no sources nor sinks of heat, and the temperature depends only on x, y, t, that is,

$$\frac{\partial T}{\partial t} = D\nabla^2 T - \boldsymbol{u} \cdot \nabla T \tag{9}$$

where $T : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ is the temperature in K, $D = 0.1\frac{\text{m}^2}{\text{s}}$ is the thermal diffusivity considered in this work, and $\boldsymbol{u} : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}^2$ the velocity field in $\frac{m}{s}$. The equation can be written as a physics loss term as

$$f_{AD}[u, v, T] = 0. \tag{10}$$

4 Finite element simulations

In order to generate data for training, validation, and testing, we simulate the PDEs using the finite element method (FEM). All simulations were performed using FEniCS (Alnæs et al., 2015). The spatial domain, $\Omega =]0, L_x[\times]0, L_y[$ with $L_x = 1$, $L_y = 1.5$, is represented by an unstructured rectangular mesh with 218 nodes and 384 triangles. For the time scheme, we take the domain $[0, T_f]$ with n uniform distributed time steps, so the discretized domain is $\{0, dt, 2dt, \ldots, ndt\}$, where $dt = \frac{T_f}{n}$. Now we give more details of the chosen values in each model.

4.1 Wave equation

We will simulate the wave equation in a rectangular domain in the time interval $]0, T_f[$, with Dirichlet boundary conditions for η:

$$\eta = 0 \text{ on } \partial\Omega \times]0, T_f[\tag{11}$$

and initial conditions on Ω

$$\begin{aligned} \eta(x,y,0) &= \exp\left(-10 \cdot \left((x-0.5)^2 + (y-0.75)^2\right)\right), \\ \frac{\partial\eta}{\partial t}(x,y,0) &= 0\,. \end{aligned} \tag{12}$$

On the other hand, the depth H is given by

$$H(x,y) = (1-x)(2-\sin(3\pi y)) \quad \text{in } \overline{\Omega}\,. \tag{13}$$

We implement FEM in this equation considering Lagrange finite elements of degree 1. The time scheme is explicit and $T_f = 1.0$, $n = 100$.

4.2 Shallow water equations

For the shallow water system we consider Dirichlet boundary conditions on all the boundary:

$$\boldsymbol{u}(x,y,t) = 0 \quad \text{on } \partial\Omega \times [0,T] \tag{14}$$

and initial conditions

$$\eta(x,y,0) = 10\exp\left(-(x-0.5)^2 - (y-0.75)^2\right) \quad \text{in } \Omega \tag{15}$$

$$\frac{\partial\eta}{\partial t}(x,y,0) = 0 \quad \text{in } \Omega \tag{16}$$

$$\boldsymbol{U}(x,y,0) = 0 \quad \text{in } \Omega \tag{17}$$

The simulations consider Lagrange finite elements of second degree. For the time scheme we used Crank-Nicholson (Allaire (2007)), and $T_f = 1$, $n = 500$.

4.3 Advection-diffusion equation

We will simulate the advection-diffusion equation in the same spatial and temporal domain as the shallow water equations but in this case we consider mixed Dirichlet and Neumann boundary conditions for the temperature (in ºC):

$$T = 0 \quad \text{on } \{0, L_x\} \times [0, L_y], \qquad \frac{\partial T}{\partial n} = 0 \quad \text{on } [0, L_x] \times \{0, L_y\}, \tag{18}$$

and initial condition (in ºC):

$$T(x,y,0) = 15 + 45 \cdot \exp\left(-(x-0.5)^2 - (y-0.75)^2\right). \tag{19}$$

We performed a θ-scheme in time with $\theta = 0.75$, and $T_f = 1$, $n = 500$. The velocity is given by the solution of the shallow water system.

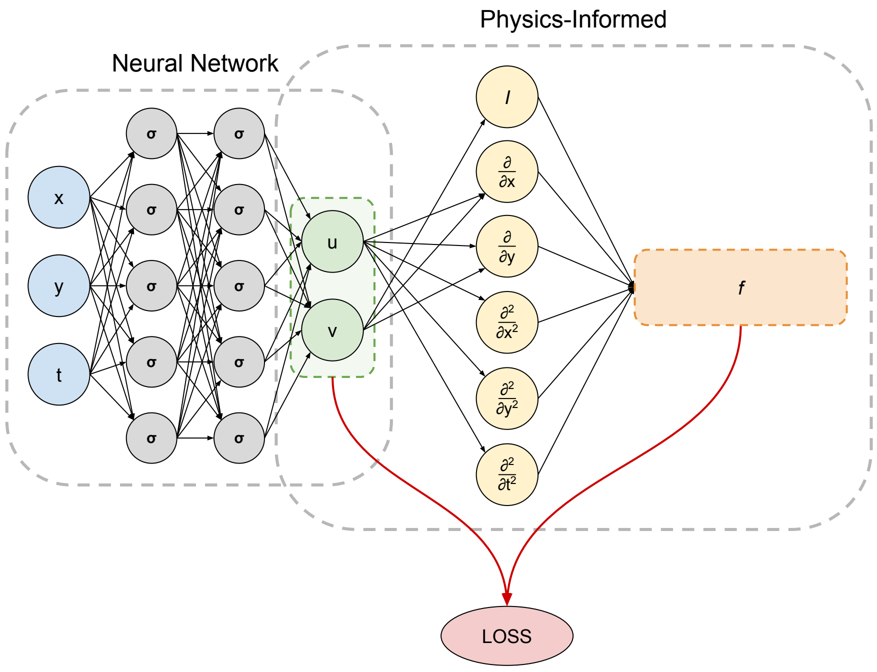

Physics-Informed
Neural Network
x
y
t
σ
u
v
I
$\frac{\partial}{\partial x}$
$\frac{\partial}{\partial y}$
$\frac{\partial^2}{\partial x^2}$
$\frac{\partial^2}{\partial y^2}$
$\frac{\partial^2}{\partial t^2}$
f
LOSS

Figure 1: Example of a physics informed neural network with inputs x, y, and t; outputs u and v; and two hidden layers. Using automatic gradient calculation we can differentiate the neural network by its input variables and construct a physics error function f. The loss function involves a loss term for the data and a loss term for the physics function.

5 Physics informed neural network

Physics informed neural networks (PINNs) are models that are trained to obey laws in physics described by (non-linear) partial differential equations (PDEs). They can be used used to solve supervised tasks in which we both minimize the error with respect to the data and to the physics law. We define the loss function as follows

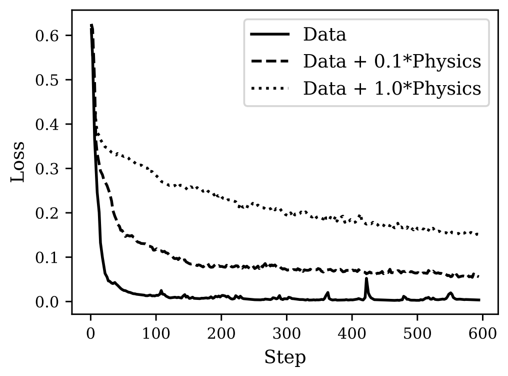
$$\mathcal{L} = \mathcal{L}_{data} + w\mathcal{L}_{physics} \tag{20}$$

where $\mathcal{L}_{data}$ is the loss with respect to the data, $\mathcal{L}_{physics}$ is the loss with respect to the PDEs of a physics law, and w is the relative weight of the physics loss. If f_i is the error of each physical condition, then using the mean-squared error for both losses we obtain

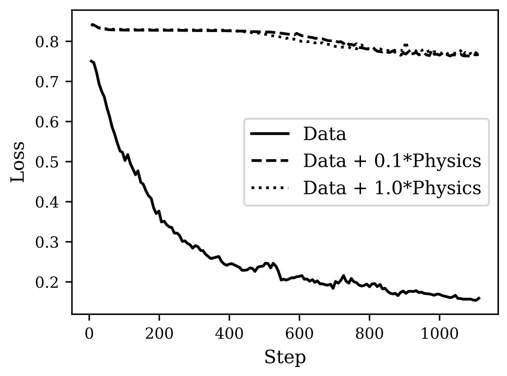
$$\mathcal{L} = \sum_i (y_i - \hat{y}_i)^2 + w\sum_i f_i^2. \tag{21}$$

Any neural network such as fully connected, recurrent, and convolutional networks can be used, where we make use of automatic gradient calculations as used by back-propagation to compute derivatives of the output variables with respect to the input variables. An example of such a network for training the wave equation is shown in Fig. 1, with f representing Eq. 2. The network is designed with input variables x, y, and t and output variables u and v. The output variables are used directly to calculate the data loss term, while for the physics loss term we differentiate the variables with respect to the input variables as needed for the physics loss function.

The objective is to train both in a supervisory manner with measured or simulated data (i.e. $\mathcal{L}_{data}$), as well as training to minimize the departure from the physics law. The $\mathcal{L}_{physics}$ term ensures that the neural network generalizes better for unseen data by preventing overfitting. The physics loss encourages that the output variables are not just trained to a local region around the input values of the given data, but that also their first and/or second-degree derivations (depending on the PDEs)



(a) Wave equation



(b) Shallow water and advection-diffusion equations

Figure 2: Validation loss of (2a) the wave equation and (2b) the shallow water and advection-diffusion equations. Both are trained three times with a physics loss weight w of 0, 0.1, and 1 using ReLU activation functions. In both instances, using solely the loss of the data $\mathcal{L}_{data}$ results in a neural network with higher accuracy after a fixed number of training iterations.

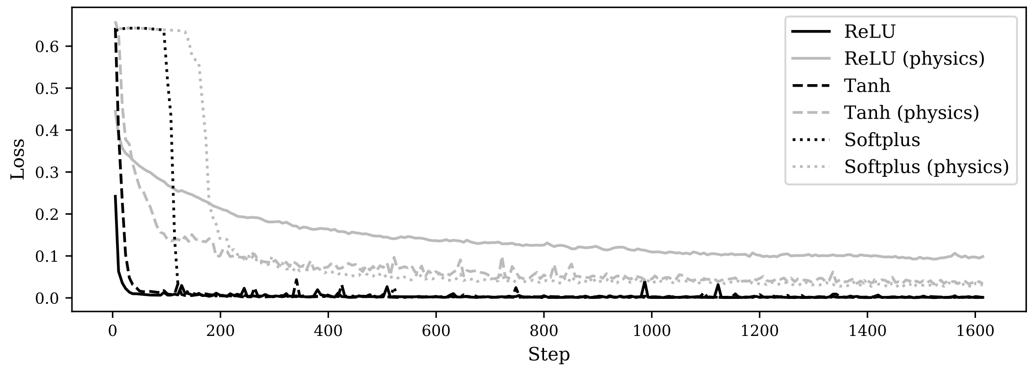


Figure 3: Validation loss of the wave equation model with different activation functions, each is trained using solely the data loss term $\mathcal{L}_{data}$ and is compared with the addition of the physics loss term $\mathcal{L}_{physics}$ using a physics loss weight $w = 1.0$. For all three cases, the loss converges much faster when training only using the data loss term.

match with our physical understanding of the model thus spurring better predictive power outside the region of the training data.

Observe that the neural network must have input variables that correspond to the physics law's derivative terms. That is if $\frac{\partial u}{\partial x}$ is part of our physics loss function then x must be an input variable and u an output variable of our neural network.

6 RESULTS

Our models are composed of five hidden layers of 100 neurons. The models are optimized using the Adam optimizer with a learning rate of $1 \cdot 10^{-3}$. Additionally, adaptive activation functions from Jagtap et al. (2020a;b) are used with an initial trainable adaptive rate of $a = 0.1$ and a fixed multiplier of $n = 10$. The data set is split into 80% training, 10% validation, and 10% test.

To improve training results, the input data is normalized to have zero mean and unit standard deviation. For the physics loss to be correctly calculated, the output of the neural network is de-normalized before calculating the PDEs. Finally, to be able to compare the training results for each of the different physics loss weights w or activation functions, we calculate the validation loss using only the data loss term $\mathcal{L}_{data}$. In case the physics loss would make the model generalize better, it should improve the loss term of the data on unseen data from the validation data set.

See Fig. 2 for the validation loss obtained by both models. It can be observed that adding a physics loss term to the loss function degrades accuracy on unseen data, thus suggesting that the model generalizes worse that with just a data loss term. In all cases the network can learn the structure of the data quicker and with higher accuracy when using just the data loss term, suggesting that the physics loss term inhibits the network from properly learning the latent structure of the data.

Additionally, in Fig. 3 the validation loss for the wave equation is shown using different activation functions. Moseley et al. (2020) advocates for the use of activation functions with non-zero second derivatives to properly learn the the second time derivative in the wave equation. We thus compare the performance of the ReLU, $\tanh$, and $\mathrm{softplus}$ activation functions, confirming worse training results in all three cases when adding a physics loss term. Additionally, it can be observed that both the $\tanh$ and $\mathrm{softplus}$ functions are slower to learn from the data than the ReLU function, but perform better than the ReLU function when adding a physics loss term.

7 Discussion

While we have found many papers stating that using a physics loss improves overall performance and generalizability of neural networks (for example, see Raissi et al. (2019); Jagtap et al. (2020a;b); Moseley et al. (2020)), we have observed the contrary. Using the same model and hyperparameters both with and without a physics loss term suggests in our case that the physics term inhibits the model from training well. We note that, contrasting with our work, Raissi et al. (2019) obtains better results by using less training data.

The neural network likely increases in complexity when imposing the physical conditions during training, as evidenced by Moseley et al. (2020) mentioning the complexity in training the wave equation. However, even when using activation functions with non-zero second derivatives as suggested by Moseley et al. (2020), such as the $\tanh$ or $\mathrm{softplus}$ functions, the model's performance is worse. The extra complexity from adding physical conditions may require extra hidden layers or wider layers to fully learn the PDEs and their second derivatives.

As the equations of interest are inherently wave-like, we propose to explore the benefits of Fourier transformations for learning the equations more effectively, as was done by Li et al. (2020). This would enable the network to learn patterns of the data in the frequency space. Additionally, to generalize the model better for unseen data, we propose the use of dropout by Hinton et al. (2012) to randomly turn off a percentage of the neurons. This forces the network to learn redundancies in case certain neurons are turned off, increasing the stability of the network, and the likelihood to generalize better for unseen data. However, for both techniques it is unclear whether adding a physics loss term would improve results.

Concluding, in this paper we have studied the behaviour of PINNs concerning the wave, shallow water, and advection-diffusion equations. We note that the added loss term for the deviation to the physical law inhibits the network from training and generalizing well to unseen data, despite using various activation functions, which may be attributed to the large amount of training data used. It is yet unclear how neural networks can learn the complexities of (second) derivatives as present in PDEs, as well as the influence of the amount of training data.

Acknowledgments

This work was funded by project CORFO 10CEII-9157 Inria Chile and Inria Challenge project OcéanIA (desc. num 14500).

References

Grégoire Allaire. *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation.* Oxford university press, 2007.

Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015. doi: 10.11588/ans.2015.100.20553.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pp. 6571–6583, 2018.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 3140–3150. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/8577-augmented-neural-odes.pdf`.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL `http://arxiv.org/abs/1207.0580`.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020a. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.109136. URL `https://www.sciencedirect.com/science/article/pii/S0021999119308411`.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2020b. doi: 10.1098/rspa.2020.0334.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2020.

Doug McLean. *Understanding aerodynamics: arguing from the real physics*. John Wiley & Sons, 2012.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Solving the wave equation with physics-informed deep learning. *arXiv preprint arXiv:2006.11894*, 2020.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL `https://www.sciencedirect.com/science/article/pii/S0021999118307125`.

A.J.C. de Saint-Venant. Theorie du mouvement non permanent des eaux, avec application aux crues des rivieres et a l'introduction de marees dans leurs lits. *Comptes rendus des seances de l'Academie des Sciences*, 36:174–154, 1871.

Nayat Sánchez-Pi, Luis Martí, André Abreu, Olivier Bernard, Colomban de Vargas, Damien Eveillard, Alejandro Maass, Pablo A. Marquet, Jacques Sainte-Marie, Julien Salomon, Marc Schoenauer, and Michèle Sebag. Artificial intelligence, machine learning and modeling for understanding the oceans and climate change. In David Dao, Evan Sherwin, Priya Donti, Lynn Kaack, Lauren Kuntz, Yumna Yusuf, David Rolnick, Catherine Nakalembe, Claire Monteleoni, and Yoshua Bengio (eds.), *Tackling Climate Change with Machine Learning workshop at NeurIPS 2020*, December 2020. URL `https://www.climatechange.ai/papers/neurips2020/93`.

Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:339–1364, 12 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.08.029.

James Johnston Stoker. *Water waves: The mathematical theory with applications*, volume 36. John Wiley & Sons, 2011.

Roger Temam. *Navier-Stokes equations: Theory and numerical analysis*, volume 343. American Mathematical Soc., 2001.